REMARKS

The Application has been carefully reviewed in light of the Office Action dated July 30, 2003 (Paper No. 6). Claims 2 to 7, 47 to 52, 54 to 71 and 92 to 96 are cancelled. Claims 1, 8 to 46, 53, 72 to 91 and 97 are in the application, of which Claims 1, 8, 27, 44 to 46, 53, 72 and 89 to 91 are the independent claims. Claims 1, 8, 44 to 46 and 89 to 91 are being amended, and Claim 97 is being added, herein. Reconsideration and further examination are respectfully requested.

Claims 1 and 46 are rejected under 35 U.S.C. § 102(b) over U.S. Patent No. 5,450,165 (Henderson) and under 35 U.S.C. § 102(e) over U.S. Patent 6,404,517 (Chao). Claims 2 to 26, 44, 47 to 71, 89 and 91 to 96 are rejected under 35 U.S.C. § 102(e) over Chao, and Claims 45 and 90 are rejected under 35 U.S.C. § 102(b) over U.S. 5,537,516 (Sherman). Reconsideration and withdrawal of these rejections are respectfully requested.

Initially, Applicants gratefully acknowledge the continued indication in the Office Action that Claims 27 to 43 and 72 to 88 recite allowable subject matter.

The present invention generally concerns a software application programming interface (API) that allows one piece of computer software, such as a color calibration software program, communicate with another piece of software to use a color measuring device. In particular, the present invention relates to a software API that provides a common software interface between a software application program and plural different types of color measuring devices, such as an spectrophotometer and a colorimeter, for example.

By virtue of this arrangement, the software application program need not include separate routines for each different type of color measuring device. Instead, the application program is able to use the common software interface provided by the software API for each of the different types of color measuring devices.

Claims 1 and 46

Turning to the particular language of the claims, Claim 1 defines computer-executable process steps to provide a software application programming interface (API). The software application programming interface provides a common software interface between an application program and a plurality of color measuring devices including a first color measuring device and a second color measuring device each having at least one color measuring sensor. The first color measuring device and the second color measuring device are different types of color measuring devices. The computer-executable process steps comprising plural functions for operating any of the plurality of color measuring devices, wherein in order to complete an operation performed by at least one of the plural functions, the function that performs the operation must be called a number of times which is different for at least the two different types of color measuring devices, and wherein for a color measuring device that is being operated, the API provides the application program with flow control data of the number of times that the function must be called.

The applied art, namely Henderson and Chao, is not seen to disclose the features of the claim. More particularly, the applied art is not seen to disclose a software application programming interface, which provides a common software interface between an application program and first and second color measuring devices which are different types of color measuring devices, wherein in order to complete an operation performed by at least one of the plural functions, the function that performs the operation must be called a number of times which is different for at least the two different types of color measuring devices, and wherein for a color measuring device that is being operated, the software API provides the application program with flow control data of the number of times that the function must be called.

The Office Action states, at page 2, that Henderson's detector 104 hardware component is the same as a software application programming interface, and that a set of densitometers are plural different types of color measuring devices. The position taken in the Office Action is not believed to be technically sufficient for at least the reasons discussed below.

Initially, reference is respectfully made to the definition of an application programming interface (API) provided at:

http://en.wikipedia.org/wiki/Application_Programming_Interface,

which is attached hereto. In part, the definition states:

> "[a]n Application Programming Interface (API) is a set of
> definitions of the ways in which one piece of computer
> software communicates with another."

The detector 104 and densitometers 24a to 24e are not seen to disclose or to suggest a software API that provides a common software interface. In addition, and as discussed below, these hardware components are not seen to in any way function as an API that has a function that is called a different number of time for different types of measuring devices, and/or an API that provides flow control data of the number of times the function must be called to an application program.

With respect to calling a function a different number of times for two different types of color measuring devices, the Office Action at pages 2 and 3 seems to consider that the description at col. 7, lines 43 to 64 of Henderson describes this feature. However, this portion of Henderson is merely seen to describe delaying activation of a densitometer until the patch is in position for detection by the densitometer in order to compensate for the speed of a photoreceptor belt 10. Delaying activation of a densitometer is clearly not the same as a calling a function of a software API a different number of times for two different types of color measuring devices in order to perform the operation of the function.

With respect to a software API providing flow control data to an application program, the Office Action at pages 2 and 3 seems to suggest that the densitometers somehow output flow control information to an application program. The densitometers in Henderson are seen to output density information. (See Henderson, col. 4, lines 47 to 54.) Neither the detector 104 or the densitometers 24a to 24e are seen to output anything remotely related to flow control to an application program. The portion of Henderson cited as describing calling a function a different number of times for different types of measuring devices, namely col. 7, lines 43 to 67, is not seen to describe outputting this information, let alone to an application program. The cited portion of Henderson is merely seen to describe a delay in activating a densitometer, whereby a timer 106 activates a densitometer after a fixed time. When activated, the densitometer outputs density information.

Accordingly, the cited portion of Henderson is not seen to disclose or to suggest outputting a number of times that a function of a software API must be called.

In addition, the Office Action at pages 2 and 3 seems to consider that Henderson describes using different types of densitometers, such that densitometer 24a and 24b are different types of densitometers, and that the delay in activating a densitometer is in some way based on the different types of densitometers. However and as described above, the description at col. 7, lines 43 to 64 of Henderson is understood to delay activation of a densitometer to accommodate the speed of photoreceptor belt 10. In addition and in order for such an interpretation (i.e., that densitometers 24a and 24b are different types of densitometers) to be supported, it would be necessary that Henderson describe using different types of densitometers in combination, which Henderson is not seen to describe. It is unclear how such a hardware configuration would operate correctly, since it would be necessary to in some manner ensure that the different types of densitometers interoperate to generate compatible density information. Henderson is not understood to address this problem. Nothing in Henderson is seen to describe a situation where different densitometers are used in combination. In addition, it is unclear how such a configuration, even if for the sake of discussion only, would disclose or suggest a function of a software API being called a different number of times for two different types of color measuring devices.

Finally and with respect to outputting the flow control data to an application program, the Office Action at pages 2 and 3 cites col. 8, lines 47 to 57. However, the cited portion of Henderson is merely seen to indicate that an application program currently available at the time of Henderson was likely to be able to output data containing test patches often enough to generate the density information needed to monitor and control the printer. Nothing in col. 8, lines 47 to 57 or col. 7, lines 43 to 64 (discussed above) is seen to disclose or to suggest outputting flow control data to an application program by a software API, the flow control data indicating the number of times a function of the software API must be called.

Chao is also not seen to disclose the claimed features. More particularly, Chao is seen to describe printing a registration mark on a color patch sheet, which is in turn used in identifying the location of patches in a color patch. (See Chao, Abstract.) Chao's image processing unit 14, consisting of a CPU 104, RAM 106, ROM 108 and temporary register set 110 as shown in Figure 4 and discussed at col. 9, lines 6 to 26, is not a software API, but is merely seen to be a computing device with memory. The image processing unit of Chao is not seen to disclose a software application programming interface providing common software interface to provide an application program with flow control data of the number of times that a function is to be called in order to complete.

At col. 3, line 56 to col. 4, line 14, Chao is seen to indicate that a device that has been calibrated in the factory may need to be re-calibrated in the field, and that there are problems in field re-calibration with locating the patterns of test patches. At col. 7, line 34 to col. 8, line 47, Chao is seen to describe a calibration image generator 56 that directs a printer to print test patterns, a registration mark processor that recognizes the registration marks, and a calibration unit that compares patterns directed to be printed with those scanned by the scanner. At col. 6, lines 41 to 61, Chao is seen to describe using a measuring device to measure test patches in device-independent color space and a scanner to measure the test patches in device-dependent color space, and generating forward and inverse functions to convert between device dependent and device independent color spaces.

Nothing in the cited portions of Chao is seen to disclose or to suggest a function used for operating a plurality of color measuring devices including a first color measuring device which is of a different type than a second color measuring device, wherein in order to complete an operation using a color measuring device the function must be called a number of times which is different for at least the two different types of color measuring devices, and wherein an API provides an application program with flow control data of the number of times that the function must be called.

Accordingly, Chao, and in particular the cited portions thereof, is not seen to disclose the features of Claim 1. Therefore, for at least the foregoing reasons, Claim 1 is believed to be in condition for allowance. Further, Applicants submit that Claim 46 is believed to be in condition for allowance for at least the same reasons.

New Claim 97

New Claim 97 is dependent on Claim 1 and has the feature that the first color measuring device is a spectrophotometer and the second color measuring device is a colorimeter. Thus and according to Claim 97, the function of the software API is called a different number of times for the spectrophotometer and the colorimeter, and the software API provides the application program with flow control data of the number of times that the function must be called, which is different for the spectrophotometer and the colorimeter.

Since the applied art is not seen to teach or to suggest those features, Claim 97 is believed to be patentable over the applied art.

Claims 8 and 53

Claim 8 defines computer-executable process steps to provide a software application programming interface (API). The software API provides a common software interface between an application program and plural different types of color measuring devices each having at least one color measuring sensor. The computer-executable process steps comprise plural functions for operating any of the plural different types of color measuring devices. A calibrate-position function calibrates a relative position of a recording medium with respect to any of the plural different types of color measuring devices. A calibrate-sensor function calibrates any of the color measuring sensors of any of the plural different types of color measuring devices. A move-to-patch function relatively positions any of the color measuring sensors and a color patch for any of the plural different types of color measuring devices, the move-to-patch function is provided with a logical color patch number by the application program. In addition, a make-measurement function makes a color measurement of the patch at which any of the color measuring sensors is relatively positioned, the make-measurement function provides the application program with a color measurement value for the color patch. In order to complete an operation performed by at least one of the plural functions, the function that performs the operation must be called a number of times which is different for at least two different types of color measuring devices, and wherein for a color measuring device that is being operated, the API provides the application program with flow control data of the number of times that the function must be called.

The applied art, namely Chao, is not seen to disclose computer-executable process steps providing a software application programming interface as a common interface between an application program and plural different types of color measuring devices, and comprising calibrate-position, calibrate-sensor, move-to-patch, and make-measurement functions, wherein in order to complete an operation performed by one of the plural functions, the function that performs the operation must be called a number of times which is different for at least two different types of color measuring devices, and wherein for a color measuring device that is being operated, the API provides the application program with flow control data of the number of times that the function must be called.

Chao is seen to concern using forward and inverse functions to convert between device dependent and device independent color spaces, and to place registration marks on a test patch sheet, which are then used to locate the test patches on a sheet of test patches. Nothing in Chao, and in particular the cited portions of Chao, is seen to disclose functions (such as calibrate-position, calibrate-sensor, move-to-patch, make-measurement functions) of a software API called to complete an operation using a color measuring device, and in order to complete an operation performed by at least one of the plural functions, the function that performs the operation must be called a number of times which is different for at least two different types of color measuring devices, and wherein for a color measuring device that is being operated, the API provides the application program with flow control data of the number of times that the function must be called.

Accordingly, Chao, and in particular the portions of Chao cited in the Office Action, is not seen to disclose the features of Claim 8. Therefore, for at least the foregoing reasons, Claim 8 is believed to be in condition for allowance. Further, Applicants submit that Claim 53 is believed to be in condition for allowance for at least the same reasons.

Claims 9 to 26 are dependent from independent Claim 8 discussed above and are therefore believed patentable for the same reasons. Because each dependent claim is also deemed to define an additional aspect of the invention, however, the individual consideration of each on its own merits is respectfully requested.

Claims 44 and 89

Claim 44 defines a software dynamically linkable library (DLL) for making color measurements with any of plural different types of color measuring devices each having at least one color measuring sensor. The software DLL comprises plural functions each of which is for operating any of the plural different types of color measuring devices. A calibrate-position function calibrates a relative position of a recording medium with respect to any of the plural different types of color measuring . The plural functions includes a calibrate-sensor function calibrates any of the color measuring sensors of any of the plural different types of color measuring devices. A move-to-patch function relatively positions any of the color measuring sensors and a color patch for any of the plural different types of color measuring devices, the move-to-patch function being provided with a logical color patch number. A make-measurement function makes a color measurement of the patch at which any of the color measuring sensors is relatively positioned, the make-measurement function providing a color measurement value for the color patch. In order to complete an operation performed by at least one of the plural functions, the function that performs the operation must be called a number of times which is different for at least two different types of color measuring devices, and wherein for a color measuring device that is being operated, the software DLL provides flow control data of the number of times that the function must be called.

Reference is made to the above discussion concerning Claim 8. For at least the same reasons, Chao, and in particular the cited portions of Chao, is not seen to disclose a software dynamically linkable library (DLL) for making color measurements with any of plural different types of color measuring devices each having at least one color measuring sensor, the software DLL comprising plural functions (such as calibrate-position, calibrate-sensor, move-to-patch, and make-measurement functions) each of which is for operating any of the plural different types of color measuring devices, wherein in order to complete an operation performed by at least one of the plural functions, the function that performs the operation must be called a number of times which is different for at least two different types of color measuring devices, and wherein for a color measuring device that is being operated, the software DLL provides flow control data of the number of times that the function must be called..

Accordingly and for at least these reasons, Claim 44 is believed to be in condition for allowance. Further, Applicants submit that Claim 89 is believed to be in condition for allowance for at least the same reasons.

Claims 45 and 90

Claim 45 defines a color calibration program, the color calibration program comprising computer-executable process steps to calibrate color fidelity of a color printer based on color measurements made by a color measuring device of color patches printed on a recording medium by the color printer. The computer-executable process steps comprising code to generate print data for the color patches. In addition, code sends the print data to the color printer so as to print the color patches on the recording medium. Code makes color measurements of the color patches printed on the recording medium using any of plural different types of color measuring devices. The code to make color measurements calls functions provided by a software application programming interface (API) that provides a common software interface to the plural different types of color measuring devices, the code to make color measurements using the common interface. In addition, code calibrates the color fidelity of the color printer based on the color measurements.

The applied art, namely Sherman, is not seen to disclose a color calibration program comprising code to call functions provided by a software application programming interface that provides a common software interface to the plural different types of color measuring devices, the code to make color measurements using the common software interface.

Sherman is seen to describe calibration of color reproduction devices in order to produce consistent colors across a variety of reproduction devices using a set of calibration curves for correcting the color output of the color reproduction device. (See Sherman, Abstract.) At col. 13, line 31 through col. 14, line 35 and col. 17, lines 20 to 47), Sherman is seen to describe calibrating a subject system (i.e., a scanner) using a reference system, which uses a densitometer or a colorimeter, to measures densities, which are then used to convert the R, G and B readings of the subject scanner to generate readings that would be obtained using the object densitometer of the reference system.

This is not seen to in any way disclose or to suggest code to call functions provided by a software application programming interface that provides a common software interface to the plural different types of color measuring devices, the code to make color measurements using the common software interface.

Accordingly and for at least these reasons, Claim 45 is believed to be in condition for allowance. Further, Applicants submit that Claim 90 is believed to be in condition for allowance for at least the same reasons.

CONCLUSION

In view of the foregoing, the entire application is believed to be in condition for allowance, and such action is respectfully requested at the Examiner's earliest convenience.

Applicants' undersigned attorney may be reached in our Costa Mesa, California office at (714) 540-8700. All correspondence should continue to be directed to our below-listed address.

Respectfully submitted,

Carole A. Quinn

Attorney for Applicants

Registration No. 39,000

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-2200
Facsimile: (212) 218-2200
CA_MAIN 72275 v 1